Attribute-Based Signatures with User-Controlled Linkability

Ali El Kaafarani[1], Liqun Chen[2], Essam Ghadafi[3], and James Davenport[1]

[1] University of Bath, UK
[2] HP Laboratories, Bristol, UK
[3] University of Bristol, UK

Abstract. In this paper, we introduce Attribute-Based Signatures with User-Controlled Linkability (ABS-UCL). Attribute-based signatures allow a signer who has enough credentials/attributes to anonymously sign a message w.r.t. some public policy revealing neither the attributes used nor his identity. User-controlled linkability is a new feature which allows a user to make some of his signatures directed at the same recipient linkable while still retaining anonymity. Such a feature is useful for many real-life applications. We give a general framework for constructing ABS-UCL and present an efficient instantiation of the construction that supports multiple attribute authorities.

Keywords. Attribute-based signatures, security definitions, user-controlled linkability.

1 Introduction

Attribute-based cryptography can play a tremendous role in providing security to cloud computing, whether for privacy/access control (encryption) or for authentication (signatures). Attribute-based encryption [21, 35] is a *natural* generalization of Identity-Based Encryption (IBE) [34, 9, 13] and its subsequent fuzzy variant [35] in the sense that it enables fine-grained control of access to encrypted data.

Attribute-Based Signatures (ABS) [27] allow a signer owning a set of attributes to sign messages w.r.t. any public access policy satisfied by his attributes revealing neither his identity nor the set of attributes used in the signing. Attribute-based signatures proved to be a powerful primitive and many existing signature-related notions such as ring signatures [33] and group signatures [10] could be viewed as special cases of attribute-based signatures. For a comparison with other primitives, we refer to [30]. The authors in [30] also showed many application of ABS including attribute-based messaging [8], trust negotiation [17] and leaking secrets.

Some constructions of ABS consider multiple authorities while others only support a single attribute authority. Okamoto et al. [32] and El Kaafarani et al. [14] provide the first schemes working in a decentralized fashion, where multiple attribute authorities are involved in the scheme, with no reliance on a central authority. To add accountability to attribute-based signatures, [25, 15, 14, 20] grant a designated tracing authority the power to revoke anonymity and reveal the identity of the signer in the case of a dispute. [20] strengthen the security notions of [14] but at the expense of having a public key infrastructure. Direct Anonymous Attestation (DAA) [5, 3] adds a new interesting feature, namely, the *user-controlled linkability* (UCL). This is a lightweight solution that avoids having a designated tracing authority, which had previously represented a bottleneck to users' privacy. In addition, it allows the user to opt to make some of his signatures directed at the same verifier linkable without sacrificing anonymity. Unlike the reliance on tracing authorities, which are generally thought of as "for trouble-shooting", UCL is intended to be built into normal use. For example, in the world of attributes, assume that a signer wants to establish a session (in a analogous way to the idea of cookies) with a recipient and maintain this session in a convincing way that he is indeed the same person whom the recipient is communicating with, not someone else who also has enough credentials to satisfy the same policy in question; the tracing authority cannot help here, whereas user-controlled linkability is an ideal functionality for such a scenario.

Existing ABS schemes differ from each other by the expressiveness of the policies they support. For instance, we have constructions supporting non-monotonic policies, e.g. [31, 15], and those supporting monotonic policies, e.g. [30], both with signatures' size linear in the length of the policy. There are also constructions supporting threshold policies, e.g. [36, 26, 23, 18], where some of them yield constant-size signatures.

Contribution. We provide security definitions and a general framework for constructing attribute-based signatures with user-controlled linkability. Instantiations of the tools used in our generic construction exist in both the random oracle [1] and the standard models. For efficiency reasons, we provide an instantiation in the random oracle model.

Paper Organization. In Section 2, we define the notion of ABS-UCL, giving its syntax along with the security definitions. In Section 3, we give the cryptographic building blocks needed for ABS-UCL. We present our general framework in Section 4, whereas in Section 5, we give a concrete construction of ABS-UCL along with the security analysis. We conclude the paper by comparing our notion to other notions in Section 6.

2 Definition and Security of ABS-UCL

In this section, we define the notion of Attribute-Based Signatures with User-Controlled Linkability (ABS-UCL), and present its security requirements. Our notion supports multiple attribute authorities, each responsible for a subset of attributes.

2.1 Syntax of ABS-UCL

In an ABS-UCL scheme, we have a set $\mathbb{AA} = \{\mathsf{AA}_i\}_{i=1}^n$ of attribute authorities, where $\mathbb{A}_i$ is the space of attributes managed by attribute authority AA_i. The universe of attributes is defined as $\mathbb{A} = \bigcup_{i=1}^n \mathbb{A}_i$. Assume that $\mathcal{A} \subset \mathbb{A}$ is a set of attributes for which a certain predicate Ω is satisfied, i.e. $\Omega(\mathcal{A}) = 1$. We have, $a \in \mathcal{A} \Rightarrow \exists \mathbb{A}_i$, s.t. $a \in \mathbb{A}_i$, so attribute a is managed by attribute authority AA_i. Below are the definitions of the algorithms used in an ABS-UCL scheme, where all algorithms (bar the first three) take as implicit input pp produced by Setup.

- $\mathsf{Setup}(1^\lambda)$: On input a security parameter, it returns public parameters pp.
- $\mathsf{AASetup}(\mathsf{aid}, \mathsf{pp})$: Is run locally by attribute authority $\mathsf{AA}_{\mathsf{aid}}$ to generate its public/secret key pair $(\mathsf{vk}_{\mathsf{AA}}, \mathsf{sk}_{\mathsf{AA}})$. The authority publishes $\mathsf{vk}_{\mathsf{AA}}$ and keeps $\mathsf{sk}_{\mathsf{AA}}$ secret.
- $\mathsf{UKeyGen}(\mathsf{id}, \mathsf{pp})$: Is run by user id to generate his personal secret key $\mathsf{sk}_{\mathsf{id}}$.
- $\mathsf{AttKeyGen}(\mathsf{id}, f(\mathsf{sk}_{\mathsf{id}}), a, \mathsf{sk}_{\mathsf{AA}})$: Is run by attribute authority AA that is responsible for the attribute a, where f is an injective one-way function, it gives the user id the secret key $\mathsf{sk}_{\mathsf{id},a}$, bound to his identity id and $f(\mathsf{sk}_{\mathsf{id}})$.
- $\mathsf{Sign}(m, \Omega, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip})$: If a user has enough attributes to satisfy the predicate Ω, i.e. $\Omega(\mathcal{A}) = 1$, then he uses the corresponding secrets keys $\mathsf{sk}_{\mathsf{id},\mathcal{A}} = \{\mathsf{sk}_{\mathsf{id},a_i}\}_{a_i \in \mathcal{A}}$ to produce a valid signature $\sigma = \{\sigma_{\mathsf{ABS}}, \sigma_{\mathsf{UCL}}\}$ on the message m and the recipient tag recip w.r.t. the predicate Ω; if $\mathsf{recip} = \perp$ then $\sigma_{\mathsf{UCL}} = \perp$.
- $\mathsf{Verify}(\sigma, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega, m, \mathsf{recip})$: Takes a signature σ on the message m and the possibly empty recipient tag recip w.r.t. a predicate Ω, the verification keys $\{\mathsf{vk}_{\mathsf{AA}_i}\}_i$ of the attribute authorities managing attributes involved in Ω, and returns 1 if the signature is valid, and 0 otherwise.
- $\mathsf{Link}(\sigma_0, m_0, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega_0, \sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \Omega_1, \mathsf{recip})$: On input two signatures, two messages, two signing policies and the verification keys of

the attribute authorities managing the attributes involved in the policies, and a recipient tag, it returns 1 if the signatures are valid on their respective messages and the same non-empty recipient tag recip (w.r.t. the respective policy), i.e. if $\mathsf{recip} \neq \perp$ and ($\sigma_{\mathsf{UCL}0} = \sigma_{\mathsf{UCL}1} \neq \perp$), and 0 otherwise.

- $\mathsf{Identify}(\sigma, m, \mathsf{recip}, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega, \mathsf{sk})$: Is only used in the security model for capturing linkability. It checks whether the valid signature σ (w.r.t. the signing policy Ω) on the message m and the *non-empty* recipient tag recip was produced by the secret key sk, outputting 0/1 accordingly.

2.2 Security Definitions

We define here the security requirements of an ABS-UCL scheme.

Correctness. This requires that signatures produced by honest users verify correctly and that signatures produced by the same user to the same valid recipient (i.e. on the same non-empty recipient tag) link.

Linkability. As specified in [37], there are two methods to support user-controlled linkability in anonymous digital signatures: In the first, a designated linking authority can determine whether or not two signatures are linked; whereas in the second method, there exists a public linking algorithm which can be run by any party. Our model supports the latter. We require that only valid signatures directed at the same recipient and which were produced by the same user link. In the game the adversary can choose all the secret keys of the users and attribute authorities. The adversary outputs $(\sigma_1, \mathsf{recip}_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega_1, \mathsf{sk}_1)$ and $(\sigma_2, \mathsf{recip}_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \Omega_2, \mathsf{sk}_2)$. It wins if σ_i is valid (w.r.t. Ω_i) on m_i and recip_i, for $i = 1, 2$ and either of the following holds:

- σ_1 was produced by sk_1 and σ_2 was produced by sk_2 where $\mathsf{sk}_1 = \mathsf{sk}_2$ and $\mathsf{recip} = \mathsf{recip}_1 = \mathsf{recip}_2 \neq \perp$ but $\mathsf{Link}(\sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega_1, \sigma_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \Omega_2, \mathsf{recip}) = 0$.
- σ_1 was produced by sk_1 and σ_2 was produced by sk_2 where $\mathsf{sk}_1 = \mathsf{sk}_2$ and $\mathsf{Link}(\sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega_1, \sigma_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \Omega_2, \mathsf{recip}_k) = 1$ for $k \in \{1, 2\}$ and either $\mathsf{recip}_k = \perp$ or $\mathsf{recip}_1 \neq \mathsf{recip}_2$.
- σ_1 was produced by sk_1 and σ_2 was produced by sk_2 where $\mathsf{sk}_1 \neq \mathsf{sk}_2$ and $\mathsf{recip} = \mathsf{recip}_1 = \mathsf{recip}_2 \neq \perp$ and $\mathsf{Link}(\sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega_1, \sigma_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \Omega_2, \mathsf{recip}) = 1$.

In summary, this requires that signatures by the same user on the same non-empty recipient tag link. Also, signatures by different users but

on the same recipient tag or those by the same user but on different recipient tags do not link.

Anonymity. This requires that a signature reveals neither the identity of the signer nor the attributes used in the signing. In the anonymity game, we have the following:

- Adversary's Capabilities: Full control over *all* attribute authorities. It can also ask for the secret keys of signers of its choice; those signers will be referred to as corrupt users. In addition, the adversary can ask for the secret key of any attribute and has a signing oracle that it can query on messages and recipient tags on behalf of honest users.
- Adversary's Challenge: The adversary outputs $(m, \mathsf{id}_0, \mathcal{A}_0, \mathsf{id}_1, \mathcal{A}_1, \Omega, \mathsf{recip})$ where $\Omega(\mathcal{A}_i) = 1$ for $i = 0, 1$. If $\mathsf{recip} \neq \perp$ then we require that throughout the game (i.e. even after the challenge phase) id_0 and id_1 must be honest (i.e. their personal secret keys are not revealed to the adversary), and that neither of $(\mathsf{id}_0, \mathsf{recip}), (\mathsf{id}_1, \mathsf{recip})$ is queried to the signing oracle. This ensures that the adversary cannot trivially win by exploiting the linkability feature.
  The adversary gets back a signature σ_b produced using $(\mathsf{id}_b, \mathcal{A}_b)$ for $b \leftarrow \{0, 1\}$. After this, the adversary can continue accessing its oracles as long as it does not violate the above two conditions.
- Adversary's Output: The adversary outputs its guess b^* and wins if $b^* = b$.

Unforgeability. This requires that users cannot output signatures on (message, recipient tag) pairs w.r.t. to a signing policy not satisfied by their set of attributes, even if they pool their attributes together, which ensures collusion-resistance. In addition, since our notion supports user-controlled linkability, we additionally require that an adversary cannot produce signatures which link to other signatures by an honest user, i.e. one whose personal secret key has not been revealed to the adversary, even if all other users and attribute authorities in the system are corrupt. Note that, unlike in DAA, e.g. [2, 3], in our notion even if a user's personal secret key is revealed, only signatures on non-empty recipient tags by the user can be traced, i.e. it is impossible to trace signatures on empty recipient tags.

In the unforgeability game, we have the following:

- Adversary's Capabilities: Access to a signing oracle. Moreover, it can corrupt any attribute authority. We refer to the non-corrupted attribute authorities as honest ones. It can also ask for the personal

secret key of any user. We refer to the non-corrupted users as honest ones. It can also ask for the secret key for any attribute.

- Winning Conditions: The adversary wins if either:
  - Adversary outputs a valid signature σ on m and recip w.r.t. Ω, where $(m, \mathsf{recip}, \Omega)$ was not queried to the signing oracle, and there exists no subset of attributes $\mathcal{A}^*$ whose keys have been revealed to the adversary or managed by corrupt attribute authorities s.t. $\Omega(\mathcal{A}^*) = 1$. In other words, $\forall \mathcal{A}^*$ s.t. $\Omega(\mathcal{A}^*) = 1$, $\exists a^* \in \mathcal{A}^*$ s.t. $\Omega(\mathcal{A}^* \setminus \{a^*\}) = 0$ and a^*'s key has never been revealed to the adversary and it is managed by an honest attribute authority.
  - Adversary outputs a tuple $(m_0, \sigma_0, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega_0, m_1, \sigma_1, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \Omega_1, \mathsf{recip} \neq \perp, \mathsf{id})$, where σ_0 is valid on m_0 and recip w.r.t. Ω_0, σ_1 is valid on m_1 and recip w.r.t. Ω_1, user id is honest, $\mathsf{Link}(\sigma_0, m_0, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \Omega_0, \sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \Omega_1, \mathsf{recip}) = 1$ and either $(\mathsf{id}, m_0, \mathsf{recip}, \Omega_0)$ or $(\mathsf{id}, m_1, \mathsf{recip}, \Omega_1)$ was not queried to the signing oracle.

    Note here the adversary has more freedom than it has in the anonymity game because it is allowed to ask for signatures by the honest user it intends to frame on any recipient tag.

3 Building Blocks

Bilinear Groups. A bilinear group is a tuple $\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of a prime order p and g_1 and g_2 generate $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The function e is a non-degenerate bilinear map $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. According to [19], prime-order bilinear groups can be categorized into three main types. We will use Type-3 where $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ are known. This type is considered to be more efficient than Type-2, and definitely more efficient than Type-1, when the latter is implemented over fields of large prime characteristic.[4]

Digital Signatures. We require a Digital Signature (DS) scheme that is correct and existentially unforgeable. In our construction realised in the ROM, we will use different variants of the full Boneh-Boyen signature scheme [6]. We refer to original full Boneh-Boyen scheme as the BB scheme, whereas we refer to its modified variant originally defined in [6],

[4] One can implement Type-1 using supersingular curves over fields of small characteristics (2 or 3), however recent records on solving DLog in these fields [22], with the help of the MOV attack [28], *ring a warning bell* to avoid using Type-1 pairings in new cryptographic applications.

and used in, e.g. [12], as the BB† scheme. Both schemes are secure under the q-SDH assumption.

Let $\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ be the description of a bilinear group and $h_1 \in \mathbb{G}_1$ is a random element. The schemes are described below:

- $\mathsf{KeyGen}(\mathcal{P})$: Choose $x, y \leftarrow \mathbb{Z}_p$, set $(X, Y) = (g_2^x, g_2^y)$. The secret key is (x, y) and the verification key is (X, Y).
- $\mathrm{BB.}\mathsf{Sign}(\mathsf{sk}, m)$: To sign $m \in \mathbb{Z}_p$, choose $r \leftarrow \mathbb{Z}_p$ such that $x+ry+m \neq 0$ and compute the signature $\sigma = g_1^{1/(x+ry+m)}$. In the BB† scheme, the signature is $\sigma = (g_1 \cdot h_1^z)^{1/(x+ry+m)}$, where the BB† signer need not know the value z.
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$: if $e(\sigma, X \cdot Y^r \cdot g_2^m) = e(g_1, g_2)$ output 1, otherwise 0. In the BB† scheme, the verification equation is $e(\sigma, X \cdot Y^r \cdot g_2^m) = e(g_1 \cdot h_1^z, g_2)$

Linkable Indistinguishable Tags. A Linkable Indistinguishable Tag (LIT) scheme [3] is similar to a Message Authentication Code (MAC) but requires different security properties. It consists of a couple of algorithms KeyGen and Tag. The former, on input a security parameter, produces a secret key sk, whereas the latter, on input a message m and the secret key, outputs a tag.

Besides correctness, the security of LIT [3] requires Linkability and f-Indistinguishability. Linkability requires that an adversary who is allowed to control both the secret key and the message cannot produce equal tags unless they are tags on the same message/key pair. Indistinguishability, which is defined w.r.t. a one-way function f of the secret key, requires that an adversary who gets $f(\mathsf{sk})$ and access to a tag oracle, cannot determine whether or not a new tag on a message of its choice was produced using the same key used by the tag oracle.

As in [3], we instantiate the LIT in the ROM with the Boneh-Lynn-Shacham (BLS) signature scheme [7]. The LIT instantiation is secure under the DDH and the discrete logarithm problems [3].

Non-Interactive Zero-Knowledge Proofs. Let R be an NP relation on pairs (x, y) with a corresponding language $\mathcal{L}_{\mathsf{R}} = \{y \mid \exists\ x\ s.t. (x, y) \in \mathsf{R}\}$. A NIZK proof system Π for a relation R is a tuple of algorithms $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Extract}, \mathsf{SimSetup}, \mathsf{SimProve})$ defined as follows: Setup outputs a reference string crs and an extraction key xk which allows for witness extraction. On input (crs, x, y), Prove outputs a proof π if $\mathsf{R}(x, y) = 1$. On input (crs, y, π), Verify outputs 1 if π is a valid proof that $y \in \mathcal{L}_{\mathsf{R}}$, and 0 otherwise. $\mathsf{Extract}$ outputs the witness x from a valid proof π. Finally, $\mathsf{SimSetup}$ outputs a simulated reference string $\mathsf{crs}_{\mathsf{sim}}$ and

a trapdoor tr, which is used by $\mathsf{SimProve}$ to simulate proofs without a witness.

We require: completeness, soundness and zero-knowledge. Completeness requires that honestly generated proofs are accepted; Soundness requires that it is infeasible to produce a convincing proof for a false statement; Zero-knowledge requires that a proof reveals no information about the witness used. For formal definitions refer to [4].

In our construction in the random oracle model, we use the Fiat–Shamir transformation [16] applied to interactive Σ-protocols.

Span Programs. A span program [24] is defined as follows:

Definition 1. *Given a monotone boolean function $\Phi : \{0,1\}^n \to \{0,1\}$, a $l \times t$ matrix M with entries in a field $\mathbb{F}$, and a labelling function $a : [l] \to [n]$ that associates M's rows to Φ's input variables. We say that M is a monotone span program for ϕ over a field $\mathbb{F}$ if for every $(x_1, \ldots, x_n) \in \{0,1\}^n$, we have the following:*

$$[\Phi(x_1, \ldots, x_n) = 1] \Leftrightarrow [\exists \boldsymbol{v} \in \mathbb{F}^{1\times t} : \boldsymbol{v} \cdot M = [1, 0, 0, \cdots, 0] \\ \wedge (\forall i : x_{a(i)} = 0 \Rightarrow v_i = 0)]$$

4 Framework for ABS with User-Controlled Linkability

Overview of the Framework.The tools we use in our generic construction are: a NIZK system Π that is sound and zero-knowledge, two existentially unforgeable signature schemes DS_1 and DS_2, a collision-resistant hash function $\mathcal{H}$ and a f-indistinguishable linkable indistinguishable tag scheme LIT. The Setup algorithm of ABS-UCL generates the common reference string crs for the NIZK system Π. It also generates a key pair $(\mathsf{vk}_{\mathsf{psdo}}, \mathsf{sk}_{\mathsf{psdo}})$ for the digital signature schemes DS_2. The public parameters of the system is set to $\mathsf{pp} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{psdo}}, \mathbb{A}, \mathcal{H})$, where $\mathbb{A}$ is the universe of attributes. For a new attribute authority to join the system, it creates a secret/verification key pair $(\mathsf{sk}_{\mathsf{aid}}, \mathsf{vk}_{\mathsf{aid}})$ for signature scheme DS_1. To generate a signing key for attribute $a \in \mathbb{A}$ for signer id, the managing attribute authority signs the signer identity along with the attribute and the image of the one-way function on his secret key, i.e. $(\mathsf{id}, a, f(\mathsf{sk}_{\mathsf{id}}))$, using $\mathsf{sk}_{\mathsf{aid}}$. The resulting signature is used as the secret key for that attribute by signer id.

To sign a message m w.r.t. a signing policy Ω, there are two cases; if the signature is linkable (i.e. on a non-empty recipient tag $\mathsf{recip} \neq \perp$), the signer first uses LIT and his secret key to compute a tag σ_{UCL} on the

recipient name recip and a NIZK proof π that such a tag verifies w.r.t. his personal secret key $\mathsf{sk_{id}}$, and that he either has a digital signature on a pseudo-attribute (following [30, 14]), i.e. the hash of the combination of the signing predicate, the message and the recipient name recip, i.e. $a_{\mathsf{psdo}} = \mathcal{H}(\Omega, m, \mathsf{recip})$, that verifies w.r.t. the verification key $\mathsf{vk_{psdo}}$ or that she has enough credentials (DS_1 signatures on $(\mathsf{id}, f(\mathsf{sk_{id}}), a_i)$) to satisfy the original signing predicate Ω. For non-linkable signatures (i.e. when $\mathsf{recip} = \perp$), it suffices to produce a NIZK proof that the signer has enough attributes to satisfy the modified predicate, i.e $\hat{\Omega} = \Omega \vee a_{\mathsf{psdo}}$, and therefore, no need for the linking part that uses LIT. Note that in this case $a_{\mathsf{psdo}} = \mathcal{H}(\Omega, m)$.

Before we define the languages for the NIZK proofs $\mathcal{L}_1$ for linkable and $\mathcal{L}_2$ for non-linkable signatures, we will generically define the format of these languages, where the secret values, aka witnesses for proofs, are underlined:

$$\mathcal{L} : \Big\{ (\text{public values} \quad \mathsf{pv}), (\text{witness} \quad \underline{\mathsf{w}}) : \mathsf{R_i}(\mathsf{pv}, \underline{\mathsf{w}}) \Big\}$$

- **Linkable signatures** ($\mathsf{recip} \neq \perp$)**:**

$$\mathcal{L}_1 : \Big\{ \big((\mathbf{vk} = \{\mathsf{vk}_i\}_{i=1}^{|\hat{\Omega}|}, \boldsymbol{a} = \{a_i\}_{i=1}^{|\hat{\Omega}|}), (\underline{\mathsf{sk_{id}}}, \underline{\mathsf{id}}, \underline{\boldsymbol{v}}, \underline{\boldsymbol{\sigma}} = \{\sigma_{a_i}\}_{i=1}^{|\hat{\Omega}|}) \big) :$$
$$\Big(\underline{\boldsymbol{v}}\mathbf{M} = [1, 0, \ldots, 0] \Big) \bigwedge_{i=1}^{|\hat{\Omega}|-1} \Big(\underline{v_i} = 0 \vee \mathsf{DS}_1.\mathsf{Verify}(\mathsf{vk}_i, \underline{\mathsf{id}}, \underline{\mathsf{sk_{id}}}, a_i, \underline{\sigma_{a_i}}) = 1 \Big)$$
$$\bigwedge \Big(\underline{v_{|\hat{\Psi}|}} = 0 \vee \mathsf{DS}_2.\mathsf{Verify}(\mathsf{vk_{psdo}}, a_{\mathsf{psdo}}, \underline{\sigma_{a_{\mathsf{psdo}}}}) = 1 \Big)$$
$$\bigwedge \Big(\mathsf{LIT}.\mathsf{Tag}(\underline{\mathsf{sk_{id}}}, \mathsf{recip}) = \sigma_{\mathsf{UCL}} \Big\}.$$

- **Non-Linkable signatures** ($\mathsf{recip} = \perp$)**:**

$$\mathcal{L}_2 : \Big\{ \big((\mathbf{vk} = \{\mathsf{vk}_i\}_{i=1}^{|\hat{\Omega}|}, \boldsymbol{a} = \{a_i\}_{i=1}^{|\hat{\Omega}|}), (\underline{\mathsf{sk_{id}}}, \underline{\mathsf{id}}, \underline{\boldsymbol{v}}, \underline{\boldsymbol{\sigma}} = \{\sigma_{a_i}\}_{i=1}^{|\hat{\Omega}|}) \big) :$$
$$\Big(\underline{\boldsymbol{v}}\mathbf{M} = [1, 0, \ldots, 0] \Big) \bigwedge_{i=1}^{|\hat{\Omega}|-1} \Big(\underline{v_i} = 0 \vee \mathsf{DS}_1.\mathsf{Verify}(\mathsf{vk}_i, \underline{\mathsf{id}}, \underline{\mathsf{sk_{id}}}, a_i, \underline{\sigma_{a_i}}) = 1 \Big)$$
$$\bigwedge \Big(\underline{v_{|\hat{\Psi}|}} = 0 \vee \mathsf{DS}_2.\mathsf{Verify}(\mathsf{vk_{psdo}}, a_{\mathsf{psdo}}, \underline{\sigma_{a_{\mathsf{psdo}}}}) = 1 \Big)$$

We use a span program (Section 3) to prove the satisfiability of the extended predicate $\hat{\Omega}$. Using a public matrix $\mathbf{M}$, the signer needs to prove the ownership of a *secret* vector $\boldsymbol{v} \in \mathbb{Z}_p^{|\hat{\Omega}|}$ for which $\boldsymbol{v}\mathbf{M} = [1, 0, \ldots, 0]$.

The zero elements in this vector $\boldsymbol{v}$ corresponds to attributes that the signer does not actually need in order to satisfy the predicate. For these values, the signer can safely choose random signatures. For the non-zero elements in $\boldsymbol{v}$, the signer needs to prove ownership of their corresponding attributes/pseudo-attribute.

The hiding property of the Π system ensures that the proof π does not reveal how the modified predicate $\hat{\Omega}$ was satisfied.

The pseudo-attribute is used for two reasons; firstly, it binds the signature to the message, the signing predicate, and the recipient name recip if the the signature is linkable. Secondly, the secret signing key $\mathsf{sk}_{\mathsf{psdo}}$ for the digital signature scheme DS will be used as a trapdoor in the security proofs to allow its holder to simulate signatures and sign on behalf of any signer without knowing their secret keys. That could be done by producing a signature on the pseudo-attribute associated with the message and the signing predicate.

The full proof for the following Theorem is in the full version.

Theorem 1. *The generic construction of the attribute-based signature with user-controlled linkability* ABS-UCL *given above is secure if the underlying building blocks are secure.*

5 A Concrete Construction of ABS-UCL

Description of the Construction. The signer's task is to provide a zero-knowledge proof of knowledge π w.r.t. the languages defined earlier, i.e. $\mathcal{L}_1$ and $\mathcal{L}_2$, depending on whether or not the signature is linkable. We instantiate DS_1 using the BB† scheme and DS_2 using the BB scheme. The proof will be made of 3 parts (or 2 if non-linkable). The first deals with the Span program to show how to hide which subset of attributes the signer has used to satisfy the modified predicate $\hat{\Omega}$. For this, the signer proves that he has used a secret vector $\boldsymbol{v}$ to span the public matrix $\mathbf{M} \in \mathbb{Z}_p^{\alpha \times \theta}$ of the span program, where $\alpha = |\hat{\Omega}|$. The second part is to show that the signatures verify correctly w.r.t. their corresponding verification keys, where the span program can safely let the signer choose random signatures for the attributes which he does not own/want to use. The third part is to show that, when the signature is supposed to be linkable, the linking part indeed uses the same user secret key used in the rest of the proof. Not that the group elements used later in the commitments, i.e. k_1, k_2 and k_3 are parts of the public parameters pp whereas sk is the signer's secret key.

Part 1: Span program

Prove that $\underline{\boldsymbol{v}}\mathbf{M} = [1, 0, \ldots, 0]$, this can be done by proving the following:

$$\sum_{i=1}^{|\hat{\Omega}|} v_i \mathbf{M}_{ij} = \begin{cases} 1 & j = 1 \\ 0 & 2 \leq j \leq \theta \end{cases} \tag{1}$$

- Commitments of vector $\boldsymbol{v}$
  - $\beta_{v_i}, \beta_{t_i}, t_i \leftarrow \mathbb{Z}_p, i = 1 \ldots \alpha.$
  - $\mathcal{V}_i = g_1^{\beta_{v_i}} \cdot k_3^{\beta_{t_i}}; \quad \hat{v_i} = g_1^{v_i} \cdot k_3^{t_i}$
- Proof of Statement
  - $\forall j \in [1, \theta]$ compute: $\Lambda_j = \prod_{i=1}^{\alpha} k_3^{t_i . M_{ij}}; \quad \lambda_j = \prod_{i=1}^{\alpha} (k_3^{M_{ij}})^{\beta_{t_i}}$

Part 2: DS_1 *and* DS_2

Now each verification equation is as follows:

$$e(\underline{\sigma_{a_i}}^{\underline{v_i}}, X \cdot Y^{\underline{r}} \cdot g_2^{a_i || \underline{\mathsf{id}}}) = e(g_1, g_2) \cdot e(h_1^{\underline{\mathsf{sk}}}, g_2)$$

DS_1 is instantiated using the BB† scheme whereas DS_2 is instantiated using the BB scheme. The signatures are as follows:

$$\sigma_{a_i} = \begin{cases} (g_1 \cdot h_1^{\mathsf{sk}})^{1/(x_i + y_i r_i + a_i || \mathsf{id})} & \text{regular attributes} \\ g_1^{1/(x_i + y_i r_i + a_{\mathsf{psdo}})} & \text{pseudo-attributes} \end{cases}$$

Where the public keys of an attribute a_i is the couple of group elements $X_i = g_2^{x_i}$ and $Y_i = g_2^{y_i}$. The identity of the signer is id and his secret key is sk. In order to use the secret vector $\boldsymbol{v}$ to hide the subset of attributes used to satisfy the predicate Ω, we can simply raise each σ_{a_i} to its corresponding vector value v_i, when v_i is zero, the signer does not want to this attribute, and therefore he can replace the signature by a random value.

- Commitments of $(\sigma_{a_i}, r_i), i \in [1, \alpha]$ and the signer identity id:
  Pick $\rho_{v_i}, \rho_{\mathsf{id}}, \rho_{r_i}, \rho_{\mathsf{sk}}, \beta_{\rho_{\mathsf{sk}}}, \beta_{\mathsf{id}\rho_{v_i}}, \beta_{r_i}, \beta_{\rho_i}, \beta_{\mathsf{id}}, \beta_{\rho_{r_i}}, \beta_{\rho_{\mathsf{id}}}, \beta_{cs}, \leftarrow \mathbb{Z}_p$, and compute:

$$T_i = {\sigma_{a_i}}^{v_i} \cdot {k_1}^{\rho_{v_i}}, \quad K_i = Y^{r_i} \cdot {k_2}^{\rho_{r_i}}, \quad Z = h_1^{\mathsf{sk}} \cdot k_1^{\rho_{\mathsf{sk}}} \quad U = {g_2}^{\mathsf{id}} \cdot {k_2}^{\rho_{\mathsf{id}}}$$

$$\hat{K}_i = Y_i^{\beta_{r_i}} \cdot {k_2}^{\beta_{\rho_{r_i}}}, \quad \hat{Z} = h_1^{\beta_{\mathsf{sk}}} \cdot k_1^{\beta_{\rho_{\mathsf{sk}}}}, \quad \hat{U} = {g_2}^{\beta_{\mathsf{id}}} \cdot {k_2}^{\beta_{\rho_{\mathsf{id}}}}$$

  Let, $\forall i \in [1, \alpha - 1] : \rho_i = \rho_{r_i} + \rho_{\mathsf{id}}$ whereas $\rho_\alpha = \rho_{r_\alpha}$.

- Simplification: (can be done by both prover and verifier)

$$X_i' = e(k_1, X_i \cdot g_2^{a_i \cdot 2^{|\mathsf{id}|}}) \quad Y_i' = e(k_1, Y_i) \quad R = e(k_1, g_2)$$
$$T_i' = e(T_i, k_2) \quad D' = e(k_1, G_2^{a_{\mathsf{psdo}}})$$

- Knowledge of Exponents
  $\forall i \in [1, \alpha]$ and $\forall j \in [1, \theta]$, compute:

$$\mathcal{X'}_{ij} = (X_i'^{M_{ij}})^{\beta_{\rho_{v_i}}} \quad \mathcal{Y'}_{ij} = (Y_i'^{M_{ij}})^{\beta_{r_i \rho_{v_i}}} \quad \mathcal{T'}_{ij} = (T_i'^{M_{ij}})^{\beta_{\rho_i}}$$

  $\forall i \in [1, \alpha - 1]$, $\forall j \in [1, \theta]$, compute:

$$\mathcal{R}_{ij} = (R^{M_{ij}})^{\beta_{\mathsf{id}\rho_{v_i}}}$$

  $\forall j \in [1, \theta]$:

  - $\mathcal{D'}_{\alpha j} = (M'^{z_{\alpha j}})^{\beta_{\rho_{v_i}}}$
  - $\mathcal{P}_j = \mathcal{X'}_{\alpha j} \cdot \mathcal{Y'}_{\alpha j} \cdot \mathcal{T'}_{\alpha j} \cdot \mathcal{D'}_{\alpha j}$
  - $\mathcal{B}_j = \mathcal{P}_j \cdot \prod_{i=1}^{\alpha-1} \mathcal{X'}_{ij} \cdot \mathcal{Y'}_{ij} \cdot \mathcal{R}_{ij} \cdot \mathcal{T'}_{ij}$

Part 3:Linkability- LIT
The signer needs to prove the following equation:

$$\mathsf{BLS.Sign}(\underline{\mathsf{sk}}, \mathsf{recip}) = \sigma_{\mathsf{UCL}}$$

If the signature is linkable, then compute:

$$\mathcal{N} = \mathcal{H}(\mathsf{recip})^{\beta_{\mathsf{sk}}}, \quad \mathcal{L} = \Big(\frac{h_1}{H(\mathsf{recip})}\Big)^{\beta_{\mathsf{sk}}} \cdot k_1^{\beta\rho_{\mathsf{sk}}} \quad \sigma_{\mathsf{UCL}} = \mathcal{H}(\mathsf{recip})^{\mathsf{sk}},$$

otherwise; $\sigma_{\mathsf{UCL}} = \perp$.
Finally, compute the challenge c:

$$c = \mathcal{H}_{\mathsf{FS}}(\underbrace{\mathcal{N}||\mathcal{L}}_{\text{if linkable}} ||\lambda_j||\mathcal{S}_i||T_i||K_i||U||\hat{K}_i||\hat{U}||\mathcal{B}_j||Z), \forall i \in [1, \alpha], \forall j \in [1, \theta].$$

- Responses
  - $s_{v_i} = \beta_{v_i} + cv_i$, $s_{t_i} = \beta_{t_i} + ct_i$, $s_{\mathsf{id}} = \beta_{\mathsf{id}} + c\mathsf{id}$, $s_{\mathsf{sk}} = \beta_{\mathsf{sk}} + c\mathsf{sk}$, $s_{\rho_{\mathsf{sk}}} = \beta_{\rho_{\mathsf{sk}}} + c\rho_{\mathsf{sk}}$, $s_{\rho_{\mathsf{id}}} = \beta_{\rho_{\mathsf{id}}} + c\rho_{\mathsf{id}}$

- $\forall i \in [1, \alpha]$:
  $s_{\rho_{v_i}} = \beta_{\rho_{v_i}} + c\rho_{v_i}$, $s_{r_i\rho_{v_i}} = \beta_{r_i\rho_{v_i}} + c(r_i\rho_{v_i})$, $s_{\rho_i} = \beta_{\rho_i} + c\rho_i$, $s_{r_i} = \beta_{r_i} + cr_i$, $s_{\rho_{r_i}} = \beta_{\rho_{r_i}} + c\rho_{r_i}$;
- $\forall i \in [1, \alpha - 1]$, compute:
  $s_{\mathsf{id}\rho_{v_i}} = \beta_{\mathsf{id}\rho_{v_i}} + c(\mathsf{id}\rho_{v_i})$

Let $\Sigma = \{s_{\rho_{v_i}}, s_{r_i\rho_{v_i}}, s_{\mathsf{id}_i}, s_{\rho_i}, s_{r_i}, s_{\rho_{r_i}}, s_{\mathsf{id}}, s_{\rho_{\mathsf{id}}}, s_{v_i}, s_{t_i}, s_{\mathsf{sk}}, s_{\rho_{\mathsf{sk}}}\}$, the signature is:

$$\sigma_{\mathsf{ABS-UCL}} = (\Sigma, c, \{\Lambda_j\}_1^\theta, \{\hat{v_i}, T_i, K_i\}_1^\alpha, U, Z, \sigma_{\mathsf{UCL}})$$

Verification

Compute:

$$\Delta_j = e(T_\alpha, (X_\alpha \cdot K_{1\alpha} \cdot G_2^{a_{\mathsf{psdo}}})^{M_{\alpha j}})$$

$$E_j = \begin{cases} \Delta_1 \cdot \prod_{i=1}^{\alpha-1} e(T_i, (X_i \cdot K_i \cdot U)^{M_{ij}}) / e(g_1, g_2) \cdot e(Z, g_2) & j = 1 \\ \Delta_j \cdot \prod_{i=1}^{\alpha-1} e(T_i, (X_i \cdot K_i \cdot U)^{M_{ij}}) & 2 \le j \le \theta \end{cases}$$

- $\hat{U} = {g_2}^{s_{\mathsf{id}}} \cdot {k_2}^{s_{\rho_{\mathsf{id}}}} \cdot U^{-c}, \quad \hat{Z} = h_1^{s_{\mathsf{sk}}} \cdot k_1^{s_{\rho_{\mathsf{sk}}}} \cdot Z^{-c}$
- $\forall i \in [1, \alpha]$:
  $\mathcal{S}_i = {g_1}^{s_{v_i}} \cdot {k_3}^{s_{t_i}} \cdot {\hat{v_i}}^{-c} \quad \hat{K}_i = {Y_i}^{s_{r_i}} \cdot {k_2}^{s_{\rho_{r_i}}} \cdot {K_i}^{-c}$
- $\forall j \in [1, \theta]$:
  - $\lambda_j = \Lambda_j^{-c} \cdot \prod_{i=1}^{\alpha} (k_3^{M_{ij}})^{s_{t_i}}$
  - $\mathcal{P}_j = ({X'}_\alpha^{M_{\alpha j}})^{s_{\kappa\alpha}} \cdot ({Y'}_\alpha^{M_{\alpha j}})^{s_{r_\alpha\kappa\alpha}} \cdot ({T'}_\alpha^{M_{\alpha j}})^{s_{\rho\alpha}} \cdot ({D'}^{M_{\alpha j}})^{s_{\kappa\alpha}}$
  - $\mathcal{B}_j = {E_j}^{-c} \cdot \mathcal{P}_j \cdot \prod_{i=1}^{\alpha-1} ({X'}_i^{M_{ij}})^{s_{\rho v_i}} \cdot ({Y_i'}^{M_{ij}})^{s_{r_i\rho v_i}} \cdot (R^{M_{ij}})^{s_{\mathsf{id}_i}} \cdot ({T_i'}^{M_{ij}})^{s_{\rho_i}}$
- For the linkablility:
  - If $\sigma_{\mathsf{UCL}} \neq \perp$, then compute:

$$\mathcal{N} = \mathcal{H}(\mathsf{recip})^{s_{\mathsf{sk}}} \cdot (\sigma_{\mathsf{UCL}})^{-c}, \quad \mathcal{L} = \Big(\frac{h_1}{H(\mathsf{recip})}\Big)^{s_{\mathsf{sk}}} \cdot k_1^{s_{\rho_{\mathsf{sk}}}} \cdot \Big(\frac{Z}{\sigma_{\mathsf{UCL}}}\Big)^{-c}$$

- Let $\hat{c} = \mathcal{H}_{\mathsf{FS}}(\underbrace{\mathcal{N}||\mathcal{L}}_{\text{if linkable}} ||\lambda_j||\mathcal{S}_i||T_i||K_i||U||\hat{K}_i||\hat{U}||\mathcal{B}_j||Z)$,
- Verify that $\hat{c} = c$ and that the following statement holds:

$$\prod_{i=1}^{\alpha} {\hat{v_i}}^{M_{ij}} = \begin{cases} g_1 \cdot \Lambda_1 & j = 1 \\ \Lambda_j & 2 \le j \le \theta \end{cases}$$

Table 1. Existing ABS schemes and their features

Scheme	Anonymity	Traceability	Decentralisation	UCL
[14, 20]	✓	✓	✓	✘
[32]	✓	✘	✓	✘
[29]	✓	✘	✘	✘
Ours	✓	✘	✓	✓

The full proof for the following Theorem is in the full version.

Theorem 2. *The construction is secure in the random oracle model if the q-SDH, DDH and Dlog assumptions hold, and the hash function $\mathcal{H}$ is collision resistant.*

6 Comparison

In Table 1, we compare the properties offered by our notion with those offered by related attribute-based signature notions. We note that the size of the signature of our concrete construction, which uses Type-3 bilinear groups is $\mathbb{G}_1^{2\cdot|\hat{\Omega}|+\theta+2} + \mathbb{G}_2^{|\hat{\Omega}|+1} + \mathbb{Z}_p^{8\cdot|\hat{\Omega}|+4}$, where θ is the number of columns in the span program matrix $\mathbf{M}$.

Our main concern in this paper was efficiency, hence the use of random oracles. There are alternative building blocks in the literature to instantiate ABS-UCL in the standard model.

Acknowledgments. We would like to thank Russell Bradford. The third author was supported by ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO and EPSRC via grant EP/H043454/1.

References

1. M. Bellare and P. Rogaway. Random oracles are practical: A Paradigm for Designing Efficient Protocols. In *ACM-CCS 1993*, ACM, pp. 62–73.
2. D. Bernhard, G. Fuchsbauer and E. Ghadafi. Efficient Signatures of Knowledge and DAA in the Standard Model. In *ACNS 2013*, Springer LNCS 7954, 518–533, 2013.
3. D. Bernhard, G. Fuchsbauer, E. Ghadafi, N.P. Smart and B. Warinschi. Anonymous attestation with user-controlled linkability. In *International Journal of Information Security*, **12(3)**, 219–249, 2013.
4. M. Blum, P. Feldman and S. Micali. Non-interactive zero-knowledge and its applications. In *STOC 1988*, 103–112, 1988.

5. E. Brickell, L. Chen and J. Li. Simplified Security Notions of Direct Anonymous Attestation and a Concrete Scheme from Pairings. In *International Journal of Information Security*, **8(5)**, 315–330, 2009.
6. D. Boneh and X. Boyen. Short Signatures Without Random Oracles. In *EUROCRYPT 2004*, Springer LNCS 3027, 56–73, 2004.
7. D. Boneh, and B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing In *Journal of Cryptology 2004*, Springer-Verlag, 297-319, 2004.
8. R. Bobba, O. Fatemieh, F. Khan, C.A. Gunter and H. Khurana. Using Attribute-Based Access Control to Enable Attribute-Based Messaging. In *ACSAC 2006*, IEEE Computer Society 3027, 403–413, 2006.
9. D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO 2001*, 213–229, 2001.
10. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT 1991*, Springer LNCS 547, 257–265, 1991.
11. L. Chen, P. Morrissey and N.P. Smart. Pairings in Trusted Computing. In *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 1-17, 2008.
12. L. Chen. A DAA Scheme Requiring Less TPM Resources. In *Lecture Notes in Computer Science* Springer Berlin Heidelberg, 350-365, 2010.
13. C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int.*, 360–363, 2001.
14. A. El Kaafarani, E. Ghadafi and D. Khader. Decentralized Traceable Attribute-Based Signatures. In *CT-RSA'14*, Springer LNCS 8366, 327-348, 2014.
15. A. Escala, J. Herranz and P. Morillo. Revocable Attribute-Based Signatures with Adaptive Security in the Standard Model. In *AFRICACRYPT 2011*, Springer LNCS 6737, 224–241, 2011.
16. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification. and signature problems. In *CRYPTO 1986*, Springer LNCS 263, 186–194, 1986.
17. K.B. Frikken, J. Li and M.J. Atallah. Trust negotiation with hidden credentials, hidden policies, and policy cycles. In *NDSS 2006*, The Internet Society, 157–172, 2006.
18. Ma. Gagné, S. Narayan and R. Safavi-Naini. Short Pairing-Efficient Threshold-Attribute-Based Signature. In *Pairing 2012*, Springer LNCS 7708, 295–313, 2012.
19. S. Galbraith, K. Paterson and N.P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, **156**, 3113–3121, 2008.
20. E. Ghadafi. Stronger Security Notions for Decentralized Traceable Attribute-Based Signatures and More Efficient Constructions. In *Cryptology ePrint Archive, Report 2014/278*, 2014.
21. V. Goyal, O. Pandey, A. Sahai and B. Waters. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *CCS 2006*, ACM ,89–98 , 2006.
22. R. Granger, T. Kleinjung and J. Zumbragel. Breaking '128-bit Secure' Supersingular Binary Curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). In *CoRR 2014*, 2014.
23. J. Herranz, F. Laguillaumie, B. Libert and C. Ráfols. Short Attribute-Based Signatures for Threshold Predicates. In *CT-RSA 2012*, Springer LNCS 7178, 51–67, 2012.
24. M. Karchmer and A. Wigderson. On span programs. In *8th IEEE Structure in Complexity Theory*, 102–111, 1993.
25. D. Khader, L. Chen and J. H. Davenport. Certificate-Free Attribute Authentication. In *Cryptography and Coding: IMACC 2009*, Springer LNCS 5921, 301–325, 2009.

26. J. Li, M. H. Au, W. Susilo, D. Xie and K. Ren. Attribute-based signature and its applications. In *ASIACCS '10*, ACM, 60-69, 2010.
27. H.K. Maji, M. Prabhakaran and M. Rosulek. Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance. In *Cryptology ePrint Archive, Report 2008/328*, `http://eprint.iacr.org/2008/328.pdf`.
28. A. Menezes, S. A. Vanstone and T. Okamoto. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. In *Transactions on Information Theory 1993*, 80-89, 1993.
29. H.K. Maji, M. Prabhakaran and M. Rosulek. Attribute-Based Signatures. In *Cryptology ePrint Archive, Report 2010/595*, `http://eprint.iacr.org/2010/595.pdf`.
30. H.K. Maji, M. Prabhakaran and M. Rosulek. Attribute-Based Signatures. In *CT-RSA 2011*, Springer LNCS 6558, 376–392, 2011.
31. T. Okamoto and K. Takashima. Efficient Attribute-Based Signatures for Non-Monotone Predicates in the Standard Model. In *PKC 2011*, Springer LNCS 6571, 35–52, ,2011.
32. T. Okamoto and K. Takashima. Decentralized Attribute-Based Signatures. In *PKC 2013*, Springer LNCS 7778, 125–142, 2012.
33. R.L. Rivest, A. Shamir and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, Springer LNCS 2248 , 552–565, 2001.
34. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, 47–53, 1984.
35. A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *EUROCRYPT 2005*, Springer LNCS 3494, 457–473, 2005.
36. S. F. Shahandashti, and R. Safavi-Naini. Threshold Attribute-Based Signatures and Their Application to Anonymous Credential Systems. In *AFRICACRYPT 2009*, Springer LNCS 5580, 198–216, 2009.
37. ISO/IEC 20008 (all parts) Information technology – Security techniques – Anonymous digital signatures, 2013.